

REMARKS

Claims 2-10, 12-20, and 22-33 are pending in the present application. Claims 6, 16, 26, and 31 have been amended herewith. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 103, Obviousness

The Examiner rejected Claims 2-10, 12-20, and 22-33 under 35 U.S.C. § 103 as being unpatentable over May et al. (US004989133) and in view of Blumenau (US006018779A). This rejection is respectfully traversed.

Claim 2 is directed to a method in a data processing system for scheduling the execution of a plurality of commands, the data processing system including an environment which executes commands concurrently, where the commands are executed without regard to a completion of execution of any other ones of the commands. The method includes the steps of selecting a plurality of commands from the environment which executes commands concurrently, and scheduling execution of the selected plurality of commands in a programming order. The scheduling step includes: (i) encapsulating a first one of the plurality of commands in a first process and encapsulating a second one of the plurality of commands in a second process; (ii) beginning processing of the first process; (iii) executing the first one of the plurality of commands in response to the beginning processing of the first process, wherein the first one of the plurality of commands executes only while the first process is executing; and (iv) beginning processing of the second process only in response to a completion of processing of the first process. This claimed method provides a technique for ensuring that processes execute, *and complete execution*, in a particular order in an environment where processes otherwise execute concurrently without regard to the status of execution of other processes. Thus, the claimed method advantageously allows for constructing branching logic and complex scripts in such a concurrent environment, as such branching logic and complex scripts cannot be constructed in an environment where the results of previously executed processes are not necessarily available to subsequent processes.

In contrast, the teachings of the cited May reference are directed to a system that allows for time-slicing a plurality of processes, such that the plurality of processes are each given a time-slice for execution, thereby allowing efficient handling of a plurality of concurrent processes while allowing time control so that no process is executed prior to its allocated scheduling time (col. 2, lines 3-8). Importantly, processes are *scheduled based upon time*, and not based upon a particular

order of execution amongst the processes themselves. Perhaps even more importantly, there is no guarantee that a previous process *completes execution* before another process is invoked, due to the stringent time-slicing requirement that is dictated by the time of a timer, and not by process completion of a previous process (col. 20, lines 29-68). This is further evidenced by the fact that a process maintains in its workspace a pointer to a next instruction to be executed when the process resumes execution after being pre-empted by another process per the time-slicing (col. 3, lines 22-29; col. 4, line 67 – col. 5, line 2; col. 9, lines 51-54). This is also evidenced by the TIME SLICE REG 80, which sets the time at which a given process must be temporarily stopped to allow other processes to be executed (col. 10, lines 63-65), and procedure StartNextProcess which deschedules the current process and selects a next runnable process (col. 11, lines 44-48).

Claim 2 expressly recites the claimed feature of “beginning processing of said second process *only in response to a completion of processing of said first process*” (emphasis added), which advantageously ensures a specific ordering of process execution *and* specific ordering of completion of process execution. Thus, and contrary to the Examiner's assertion regarding the teaching of May, none of the cited references teach or otherwise suggest the claimed feature of “beginning processing of said second process only in response to a completion of processing of said first process”. Thus, a prima facie case of obviousness has not been established with respect to Claim 2¹, and accordingly the burden has not shifted to Applicants to rebut the obviousness assertion². In addition, as the Examiner has failed to establish a prima facie case of obviousness, the rejection of Claim 2 under 35 U.S.C. 103 is improper³.

Still further with respect to Claim 2, such claim recites features of “encapsulating said first one of said plurality of commands in a first process and encapsulating said second one of said plurality of commands in a second process; beginning processing of said first process; executing said first one of said plurality of commands in response to said beginning processing of said first process, wherein said first one of said plurality of commands executes only while said first

¹ To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. See also, *In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

² In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *Id.*

³ If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

process is executing; and beginning processing of said second process only in response to a completion of processing of said first process". In rejecting this aspect of Claim 2, the Examiner acknowledges that the cited May reference does not teach such command encapsulation in a process, but states that Blumenau teaches encapsulating a *plurality* of commands within a *single* command and executing them, and that even though Blumenau expressed that the encapsulations of a plurality of commands into a single command, it can be inferred that a single command can also be encapsulated within a command to be executed. Applicants urge that this is classic hindsight analysis, where the Examiner is using Applicants' own patent specification as a blueprint to modify the teachings of the cited references in accordance with the claimed invention. Blumenau expressly states that a *plurality* of commands are encapsulated in a *single* command to reduce processing time, and in particular to reduce processing overhead associated with arbitration that is required when executing a command across a SCSI bus (col. 2, lines 21-31). It makes no sense to infer from the teachings of Blumenau that a single command can also be encapsulated within a command to be executed, as the entire, expressed purpose of Blumenau – to reduce overhead arbitration associated with processing individual commands – would be defeated as only single commands would be sent across the SCSI bus at a time (with their associated arbitration overhead), and in fact this single command encapsulation would itself introduce additional processing (the additional processing being the encapsulation of a single command within a single command) without any associated benefit. Thus, a person of ordinary skill in the art would not have been motivated to modify the teachings of Blumenau in accordance with the features recited in Claim 2. This further evidences that the only motivation to modify the teachings of the cited references in accordance with present invention must be coming from the present invention itself, which is improper hindsight analysis.

Still further with respect to Claim 2, the cited May reference teaches an elaborate time-slicing scheme where processes share processing of single processor for certain slices of time, and then are descheduled and subsequently rescheduled after their time-slice time has ended (col. 1, lines 51-53; col. 10, lines 62-65). An elaborate memory map is established for each of the processes, with a linked list used to interlink such processes (col. 9, lines 51-68). It is not seen how this tightly-coupled process interlinking could be modified to include encapsulating one command in a first process and encapsulating another command in another process, as the inter-process linkage using linked-lists would no longer be operable due to the outer, encapsulating layer which would isolate the details of the internal processes from one another. When an obviousness

determination is based on multiple prior art references, there must be a showing of some "teaching, suggestion, or reason" to combine the references. "...absence of such suggestion to combine is dispositive in an obviousness determination". *Gambro Lundia AB v. Baxter Healthcare Corp.*, 110 F.3d 1573, 42 USPQ2d 1378 (Fed. Cir. 1997). There is simply no reason or other motivation to combine such references, as the use of encapsulated commands, as taught by Blumenau, is not compatible with the May architecture and associated process linked-list. This further establishes that Claim 2 has been erroneously rejected under 35 USC 103, as there is no teaching, suggestion or other motivation to combine such references.

Still further with respect to Claim 2, Applicants urge that the two references being used to reject all pending claims are non-analogous art, and thus a person of ordinary skill in the art would not have been motivated to combine such dissimilar teachings. May is directed to a system for executing, scheduling, and selectively linking time depend processes based upon a scheduling time, whereas Blumenau is directed to a technique for communicating with a storage device across a SCSI interface. One of ordinary skill in the intra-processor concurrent command execution art, faced with a problem associated with sequentially scheduling of processes in a concurrent process execution environment, would not look to solutions that others faced with DASD storage device communication techniques, and therefore the references are non-analogous art, per MPEP 2141.01(a). Thus, Claim 2 is further shown to have been erroneously rejected, as the art being used in such rejection is non-analogous.

Applicants initially traverse the rejection of Claims 3-10 and 31 for reasons given above with respect to Claim 2 (of which Claims 3-10 and 31 depend upon).

Further with respect to Claim 6 (and dependent Claims 7-10), Applicants urge that none of the cited references teach or suggest the claimed feature of "establishing a return code variable for the first process; and utilizing said return code variable to indicate whether said first process is currently executing". In rejecting this claim, the Examiner cites extensive, unrelated passages as teaching this claimed feature. Applicants urge that none of such cited passages teach use of a process return code variable to indicate whether a process is currently executing. For example, the Examiner cites May col. 1, line 51 - col. 3, line 28. While this passage describes a summary/overview of the May teachings, there is no specific teaching of any use of a process return code variable, for use in indicating whether a process is currently executing, or any other use. The passage cited at col. 7, 43-45 is indeed a variable, but is not used to indicate whether a process is

currently executing, but instead is used to signal, when set, that the current process should be descheduled on completion of the current instruction. Importantly, if this flag is not set, the process may still be executing, so use of this flag does not provide an affirmative indication as to whether a process is executing, but instead signals that a process should be descheduled. The passage cited beginning at col. 9 is with respect to memory allocation for various processes, and is depicted in Figure 4. Figure 4 shows various general purpose variables for a given process, with no details of how such general purpose variables are used. Figure 4 also shows variables IPTR, LINK, STATE, TLINK, and TIME. The function of these variables is described at col. 4, lines 45-68, and none of these descriptions teach that any of these variables are a return code variable for a process that is used to indicate whether a process is currently executing. Nor does the cited passage from the Blumenau reference overcome this teaching deficiency. Such passage merely describes encapsulating of a plurality of commands into a single command that is sent to a storage device. There is no mention of any type of process return code variable, for use in indicating whether a process is currently executing. Thus, it is further urged that Claim 6 (and dependent Claims 7-10) is not obvious in view of the cited references.

Still further, because May teaches that processes either run to completion or until pre-empted by a timer (per the time-slicing), there would have been no reason to determine whether a given process is executing. In contrast, as the present invention begins execution of a second process *only after completion of a first process*, there is a need to determine whether a given process is executing.

Still further with respect to Claim 7 (and dependent Claims 8-10), such claim recites use of process identifier to determine whether a process is executing. This claim depends upon Claim 6, which recites use of a process return code variable to indicate whether a first process is executing. The Examiner cites the identical passages that were recited in rejecting Claim 6 in the rejection of Claim 7. Applicants urge that Claim 7, in combination with Claim 6, recites use of a process identifier to determine if a process is executing, and use of a process return code variable to indicate that a process is executing. None of the cited references teach or otherwise suggest use of *both a process return code variable and a process identifier* for determining and indicating process execution. Thus, it is further urged that Claim 7 (and dependent Claims 8-10) is not obvious in view of the cited references.

Further with respect to Claim 8 (and dependent Claims 9 and 10), none of the cited references teach or suggest the claimed steps of "searching a process table for said first process

identifier" and "determining that said first process is executing in response to locating said process identifier in said process table", nor has the Examiner even alleged any such teaching or suggestion of a process table, or searching a process table. Thus, the Examiner has failed to establish, or even allege, a prima facie showing of obvious with respect to Claim 8. Thus, Claim 8 (and dependent Claims 9 and 10) is further shown to have been erroneously rejected under 35 U.S.C. 103.

Further with respect to Claim 9 (and dependent Claim 10), none of the cited references teach or suggest the claimed steps of "setting said return code variable equal to a first value while said first process is executing" and "setting said return code variable equal to a second value when said first process has completed executing", nor has the Examiner even alleged any such teaching or suggestion of setting a return code variable to one of two different values to indicate, respectively, process execution and process execution completion. Thus, the Examiner has failed to establish, or even allege, a prima facie showing of obvious with respect to Claim 9. Thus, Claim 9 (and dependent Claim 10) is further shown to have been erroneously rejected under 35 U.S.C. 103.

Further with respect to Claim 10, such claim is directed to use of a timer to assist in testing the process return code variable. In rejecting Claim 10, the Examiner cites the identical passages that were recited in rejecting Claim 6 in the rejection of Claim 10. Applicants urge this broad brushed approach to claim rejection fails to comply with 37 CFR 1.104(c)(2), as the pertinence of these numerous cited passages is not explained. For example, the cited Blumenau passage used in rejecting Claim 10 makes no mention whatsoever of any timer for any use at all, so it is not seen how this cited passage teaches or otherwise suggests the specific features recited in Claim 10, *each of which recite use of a timer*. While May teaches use of a timer, this timer is used for time-slicing of tasks, where a running task may be pre-empted by another task based on the time the task is set to be executed. This timer is not used to determine when to test a process return code variable, but rather is used to preempt execution of a process. Thus, Claim 10 is further shown to not be obvious in view of the cited references, as there are claimed features not taught or suggested by the cited references.

Applicants initially traverse the rejection of Claims 12-20 and 32 for similar reasons to those given above with respect to Claim 2.

Applicants further traverse the rejection of Claim 16 (and dependent Claims 17-20) for similar reasons to the further reasons given above with respect to Claim 6.

Applicants further traverse the rejection of Claim 17 (and dependent Claims 18-20) for similar reasons to the further reasons given above with respect to Claim 7.

Applicants further traverse the rejection of Claim 18 (and dependent Claims 19-20) for similar reasons to the further reasons given above with respect to Claim 8.

Applicants further traverse the rejection of Claim 19 (and dependent Claim 20) for similar reasons to the further reasons given above with respect to Claim 9.

Applicants further traverse the rejection of Claim 20 for similar reasons to the further reasons given above with respect to Claim 10.

Applicants initially traverse the rejection of Claims 22-30 and 33 for similar reasons to those given above with respect to Claim 2.

Applicants further traverse the rejection of Claim 26 (and dependent Claims 27-30) for similar reasons to the further reasons given above with respect to Claim 6.

Applicants further traverse the rejection of Claim 27 (and dependent Claims 28-30) for similar reasons to the further reasons given above with respect to Claim 7.

Applicants further traverse the rejection of Claim 28 (and dependent Claims 29-30) for similar reasons to the further reasons given above with respect to Claim 8.

Applicants further traverse the rejection of Claim 29 (and dependent Claim 30) for similar reasons to the further reasons given above with respect to Claim 9.

Applicants further traverse the rejection of Claim 30 for similar reasons to the further reasons given above with respect to Claim 10.

Further with respect to Claim 31 (and similarly for Claims 32 and 33), none of the cited references teach or suggest the claimed feature of "wherein said first process and said second process are included in a script", nor has the Examiner even alleged any such teaching or suggestion. Thus, the Examiner has failed to establish, or even allege, a prima facie showing of obvious with respect to Claim 31. Thus, Claim 31 (and similarly for Claims 32 and 33) is further shown to have been erroneously rejected under 35 U.S.C. 103.

Therefore, the rejection of Claims 2-10, 12-20, and 22-33 under 35 U.S.C. § 103 has been overcome.

II. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 9/16/05

Respectfully submitted,



Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorneys for Applicants